

# Num

用于数值；

num数据类型的值可以为：

整数； -5

小数； 3.45

其亦可呈指数地写入，例如：

$2E3$  ( $=2*10^3=2000$ ) ，

$2.5E-2$  ( $=0.025$ ) 。

始终将-8388607与+8388608之间的整数作为准确的整数储存。

```
VAR num reg1;
```

```
reg1 := 3;
```

将reg1指定为值3。

```
a := 10 DIV 3;
```

```
b := 10 MOD 3;
```

整数除法，向 a 分配  
一个整数（=3），  
并向b分配余数（=1）

# Dnum

`dnum`双值数用于数值其可以处理大于数据`num`的整数值，其使用与`num`相同。

`dnum`数据类型的值可以为：

一个整数，例如-5

一个小数，例如3.45

其亦可呈指数地写入

$$2E3 \quad (=2*10^3=2000)$$

$$2.5E-2 \quad (=0.025)$$

始终将-4503599627370496与  
+4503599627370496之间的整数作  
为准确的整数储存。

# Add

Add用于从数值变量或者永久数据对象增减一个数值。

例 1

Add B, 3;

将3增加到B，即 $B:=B+3$ 。

例 2

Add reg1, -reg2;

reg1的值减去reg2，即， $reg1:=reg1-reg2$ 。

```
VAR dnum mydnum:=5;
```

```
Add mydnum, 500000000;
```

将500000000增加到mydnum,

即mydnum:=mydnum+500000000。

# 变元

Add Name | Dname AddValue | AddDvalue

## Name

数据类型: **num**

待改变变量或者永久数据对象的名称。

## Dname

数据类型: **dnum**

待改变变量或者永久数据对象的名称。

## AddValue

数据类型: **num**

有待增加的值。

## AddDvalue

数据类型: **dnum**

有待增加的值。

```
VAR dnum mydnum:=5000;
```

```
VAR num mynum:=6000;
```

```
Add mynum, DnumToNum(mydnum \Integer);
```

将5000增加到mynum，即，`mynum:=mynum+5000`。

必须使用DnumToNum以获得num数值，该数值可以与num变量mynum一同使用。

如果有待增加的值是dnum型，且应当更改的变量/永久数据对象是num，则将产生一个运行时错误。

# Clear

用于清除数值变量或永久数据对象，即，将数值设置为0。

```
Clear reg1;
```

Reg1得以清除，即，reg1:=0。

---

```
CVAR dnum mydnum:=5;
```

```
Clear mydnum;
```

mydnum得以清除，即mydnum:=0。

# 变元

Clear Name | Dname

**Name**

数据类型: num

待清除变量或者永久数据对象的名称。

**Dname**

数据类型: dnum

待清除变量或者永久数据对象的名称。

# Incr

Incr用于向数值变量或者永久数据对象增加1。

```
Incr reg1;
```

将1增加至reg1，即 $\text{reg1} := \text{reg1} + 1$ 。

# 变元

Incr Name | Dname

**Name**

数据类型： num

待改变变量或者永久数据对象的名称。

**Dname**

数据类型： dnum

待改变变量或者永久数据对象的名称。

```
VAR num no_of_parts:=0;  
...  
WHILE stop_production=0 DO  
  
produce_part;  
  
Incr no_of_parts;  
  
TPWrite "No of produced parts= "\Num:=no_of_parts;  
  
ENDWHILE
```

更新FlexPendant示教器上各循环所产生的零件数。只要未设置输入信号stop\_production，则继续进行生产。

# Decr

**Decr**用于从数值变量或者永久数据对象减去1。

```
Decr reg1;
```

从reg1中减去1，即 $\text{reg1} := \text{reg1} - 1$ 。

# 变元

## Decr Name | Dname

### Name

数据类型: num

待缩减变量或者永久数据对象的名称。

### Dname

数据类型: dnum

待缩减变量或者永久数据对象的名称。

```
VAR num no_of_parts:=0;
```

```
...
```

```
TPReadNum no_of_parts, "How many parts should be  
produced? ";
```

```
WHILE no_of_parts>0 DO
```

```
produce_part;
```

```
Decr no_of_parts;
```

```
ENDWHILE
```

要求操作员输入待生产零件的数量。变量no\_of\_parts用于统计必须继续生产的数量。